

3層C/Sシステム開発方法論の評価実験

畑 恵介 高田 信一 山本 修一郎
NTTソフトウェア研究所
ソフトウェア技術研究部

本稿では、我々が検討を行った3層C/Sシステム開発方法論に関して、その有効性と問題点を明らかにするために実施した、評価実験の結果について報告する。評価実験の結果をもとに、3層C/Sシステム開発方法論を用いて、システムを開発した場合に作成されるドキュメントに対する評価を行った。また、評価実験結果から明らかになった、機能を各層に切り分ける際の具体的な指針、インタフェースを共通化に関する際の指針、3層C/Sシステム開発方法論を用いてシステムを開発する場合に必要な工数の指標値について報告する。

An Experiment on the Three-Tiered Client-Server System Development Methodology

Keisuke Hata, Shin-ichi Takata, Shuichiro Yamamoto
NTT Software Laboratories
Software Engineering Laboratory

In this paper, we report the result of an experiment to find out the merits and demerits of the three-tiered client-server system development methodology. We evaluate the effectiveness of the methodology in the view point of the design documentation. And we report the guidelines on how to divide the function into each tier, how to unify the interfaces which have similar function, and how much time it takes in each process to develop the system by the methodology.

1.はじめに

我々は、3層C/Sアーキテクチャ^[1]にもとづいたシステム開発方法論の検討を行っている。3層C/Sアーキテクチャでは、アプリケーション機能を、プレゼンテーション層、機能層、データ層の3つの論理的な独立した層に分離する。このように各機能を3つの層に分離する事により、業務ロジック

の変更や、GUIツール、DBMS等のソフトウェア技術の進歩、ハードウェア技術の進化が発生した場合でも、これらによる影響は各層に局所化されるため、容易に対応する事が可能である。このため、3層C/Sアーキテクチャは、今日激しく変化するビジネス環境や技術に対応できるアーキテクチャとして、注目されている。本稿では、我々が提案する、3層C/Sシステム開発方法論^{[2][3]}に対して実施

した評価実験の結果について報告する。我々のグループでは、この評価実験に関して、予備実験、製造実験、改造実験、機能拡張実験の4段階の実験を予定している。予備実験では、簡単なシステムを製造しながら方法論を検証し、これからの評価実験のための準備をする。製造実験では、方法論を用いて小規模なシステムを開発し、方法論の有効性を評価する。改造実験・機能拡張実験では、それぞれシステムの改造・機能拡張を行い、改造・機能拡張の面から方法論の有効性を評価する。本稿では、予備実験の結果について報告する。

2.3 層C/Sアーキテクチャ

3層C/Sアーキテクチャでは、アプリケーション機能を、プレゼンテーション層(Presentation layer：以下P層と呼ぶ)、機能層(Functionality layer：以下F層と呼ぶ)、データ層(Data layer：以下D層と呼ぶ)の三つの論理的な独立した層に分離する。プレゼンテーション層では、人間とコンピュータ間の入出力制御を提供する。機能層では、業務内容に依存した処理機能や、他の2つの層間の橋渡し機能を提供する。また、データ層では、データを効率的に蓄積し参照できる機能と、他の既存システムと接続する機能を提供する。このように、各機能を3つの論理的な層に分離する事により、業務ロジックの変更や、GUI ツール、DBMS等のソフトウェア技術の進歩、ハードウェア技術の進化が発生した場合でも、これらによる影響は各層に局所化されるため、容易にかつ迅速にシステムを改造できる。

3.3 層C/Sシステム開発方法論の概要

我々が提案する3層C/Sシステム開発方法論の概要について述べる。

3.1.工程

3層C/Sシステム開発方法論では、作業工程をおおまかに、前工程、3層設計工程、後工程の3つ

の工程に分類することができる。前工程では、要求分析を行い、業務フロー、データモデル、画面設計図を作成する。3層設計工程では、3層アーキテクチャを採用するうえでオリジナルな工程であり、後で述べるシナリオフロー図と、各層間のインタフェース仕様を作成する。後工程では、前の2つの工程の成果物を元に、モジュール設計を行い、コーディング、試験を行う。

3.2.前工程

前工程は、要求分析を行い、以下の物を作成する。

- 業務フローの作成：システム化する対象業務を分析し、各業務の内容と業務相互の関係を定義する。
- データモデルの作成：システム化する対象業務が扱うデータをモデル化する。
- 画面設計：業務で使用する画面の入出力、レイアウト、遷移、および帳票を定義する。
- 既存システム、既存データベースとのインタフェース抽出：開発するシステムがアクセスする既存のシステム、データベースとのインタフェースを明確にする。

3.3.3層設計工程

ユーザが画面上でボタンを押すなどのアクションを行う事により、さまざまなイベントが発生する。これらのイベントのうち、そのイベントに対するシステムの処理で、データベースへのアクセスがあるものを機能イベントと呼ぶことにする。

3層設計工程では、前工程で明らかになった情報をもとにイベントを抽出し、機能イベントに関してシナリオフロー図を作成する。シナリオフロー図の例を図1に示す。

この工程では、まず、機能を3つの層に分類する。人間とシステムのやり取りを行う機能はP層に配置する。D層にはデータを蓄積したり参照する機能と、既存システムと接続する機能を配置する。これら以外の業務ロジックに関するような機能は、すべてF

層に配置する。各層に機能を分類できたら、各層間のインタフェースを決定する。これらの情報をシナリオフロー図に記述する。

次に、このシナリオフロー図を元に、共通インタフェースの抽出を行う。各層において、同じようなインタフェースを持つ機能は共通機能として共通化する。インタフェースの共通化が終了したら、すべての機能について、インタフェース仕様書を作成する。

3.4.後工程

後工程では、前の2つの工程の成果物を元にモジュール設計を行い、コーディング、試験を行う。

4.予備実験

4.1.実験の目的

今回の予備実験は以下の点を目的として行った。

- **開発方法論の問題点の抽出:**あらかじめ検討された方法論の問題点を抽出し、その解決策を検討する。
- **ツールの習熟:**今後の評価実験に用いられるツールに習熟する。
- **指標値の抽出:**今後の評価実験での開発工数見積りのためのおおまかな指標値を抽出する。

4.2.システム構成

予備実験では以下のソフトウェアを使用した。

- ORACLE7
- VGUIDE / VGLink¹
- Visual Basic 2.0J

システムおよびソフトウェア構成を図2に示す。

4.3.システム仕様

予備実験では、「ソフトウェア開発プロジェクト進捗管理システム」の要求仕様を作成し、これを3層C/Sシステム開発方法論に従って開発した。以下にその仕様の概要について述べる。

システム開発工程管理ツール DEWAM² の概要

DEWAM は、ソフトウェア開発プロジェクトの管理を目的としたツールである。

- **プロジェクトの組織構成とメンバの管理:**ソフトウェア開発プロジェクト内の組織構成と、作業メンバを登録し、管理する機能。
- **作業計画の管理:**作業計画をあらかじめ登録し、進捗の遅れや手戻りが発生した場合には計画を変更できる機能。
- **作業成果管理:**作業者が日々の作業実績を登録し、管理者が進捗管理をするための機能。
- **作業週報作成:**毎週行われるプロジェクト内打ち合わせにおいて、グループリーダーが作業報告するための資料を作成する機能。

4.4.実験の進め方

予備実験は、方法論の問題点と、指標値を抽出するために、下記の要領で行った。

- **方法論の問題点の抽出:**方法論に沿って開発を行い、作業者が方法論の問題を発見した場合、管理者に随時報告する。管理者は問題が発生した場合メンバ全員を招集し、ミーティングを開催する。ミーティングでは、問題点について検討し、解決策を考える。問題が解決できた場合は、それを新しく方法論に盛り込む。解決が困難と判断された場合は、保留とし、予備実験終了後に改めて検討を行う。
- **指標値の抽出:**作業者の各作業項目に対する毎日の作業時間を日報で管理者に報告する。

¹ NITTソフトウェア社が提供する分散システム開発環境。文献 [4], [5] を参照

² DEWAM: System DEvelopment Work Activity Management

顧客個人データ検索業務

| UI | P層処理 | P/F I | F層処理 | F/D I | D層処理 | データ |
|--------------------------------------|---|------------------------------|--|------------------------------|--|-------|
| 個人データ 照会 ユーザ ID | ○入力チェック <IF>(OK) <ELS> ○EXIT [NG] ◎個人データ 取得 | 個人データ 取得 ユーザ ID | ○個人データ 取得 <IF>(OK) ○情報返却 ○EXIT[OK] <IF>(NF) ○EXIT[NF] <IF>(FE) ○EXIT[FE] | 個人データ 取得 ユーザ ID | ○個人データ 取得 <IF>(OK) ○情報編集 ○EXIT[OK] <IF>(NF) ○EXIT[NF] <IF>(FE) ○EXIT[FE] | 個人データ |
| 個人情報 [OK] [NG] [NF] [FE] | <IF>(OK) <ELS> ○EXIT ○個人データ 表示 | 個人情報 [OK] [NF] [FE] | | 個人情報 [OK] [NF] [FE] | | |

図1 シナリオフロー図の例

図は「顧客個人データ検索業務」のシナリオフロー図の例。シナリオフロー図には、ユーザインタフェース (UI)、P層処理、PFインタフェース (P/F I)、F層処理、FDインタフェース (F/D I)、D層処理、データの各カラムがある。それぞれのカラムに、P層処理を呼び出すためのユーザインタフェース、P層処理概要、F層処理を呼び出すためのPFインタフェース、F層処理概要、D層を呼び出すためのFDインタフェース、D層処理概要を記述する。最後のデータの各カラムには、アクセスするデータの実体名または、外部システム名を記述する。各インタフェース中で、最初の行はインタフェース名を記述し、その下の上部には入力データ、下部には出力データを記述する。

□xxx は入出力データ、□xxx は複数件の入出力データ、[]はリターンコードを表す。各層では、処理の概要を記号を用いて記述する。○は通常処理、◎は下位層呼び出し、<>は分岐、()は分岐条件、をを表す。

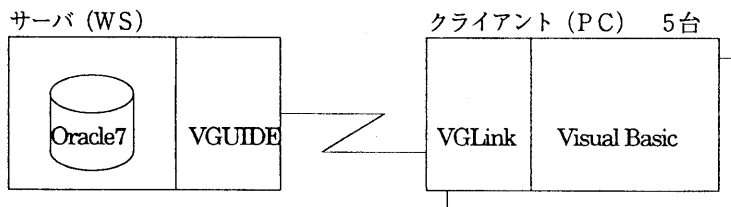


図2 システム構成

4.5.実験結果

4.5.1.開発規模

予備実験では、実験期間とのかねあいから、DEWAMの要求仕様のすべての機能について、3層設計工程までの設計は行ったが、後工程でのコーディング対象は一部の機能に限定して行った。表1、2にその開発規模を示す。

表1：要求仕様におけるシステム規模

| | |
|-----|------|
| 業務数 | 9業務 |
| 画面数 | 11画面 |
| 帳票数 | 10帳票 |

表2：コーディング対象

| | |
|-----|-----|
| 画面数 | 7画面 |
| 帳票数 | 1帳票 |

4.5.2.生産物

予備実験で開発したシステムの規模は、約3Kラインであった。表3、4にその詳細を示す。

表3：仕様書

| | |
|-----------|--------|
| 要求仕様書 | 44ページ |
| シナリオフロー図 | 49 |
| インタフェース仕様 | 145 |
| データベース仕様 | 5ページ |
| その他 | 約10ページ |

表4：コード

| | |
|-----------|---------|
| 画面(Form)数 | 7 |
| P層モジュール数 | 20 |
| F層モジュール数 | 21 |
| D層モジュール数 | 24 |
| P層ライン数 | 707ライン |
| F層ライン数 | 836ライン |
| D層ライン数 | 1309ライン |

※P層/F層はすべてVisualBasicで記述。

D層は約半分はVisualBasicで、残りはVGUIDEのCP言語で記述。

4.5.3.開発工数

予備実験に要した開発工数を各工程別に示す(表5)。

表5：工程別の工数

| | |
|--------|------|
| 前工程 | 8人日 |
| 3層設計工程 | 32人日 |
| 後工程 | 30人日 |
| 合計 | 70人日 |

予備実験では、平均5個の機能イベントを有する画面1つに対して、前工程から後工程までの全ての開発工数が平均約10人日であった。ただし、この値には今回初めて行った開発のため、ツール習熟や環境に慣れるための時間などが含まれる。

5.方法論の評価

3層C/Sシステム開発方法論では、3層設計工程が従来の開発手法と異なるため、3層設計工程について評価を行った。

5.1.問題点と解決策

予備実験を行った結果、あらかじめ検討された方法論に、いくつかの不備な点が存在する事が明らかになった。以下にその問題点と、問題点について検討した解決策について述べる。

問題点1 シナリオフロー図を書く時の図1枚の単位はどのように決定するか。

シナリオフロー図を記述する1枚の単位は、当初、画面から発生する機能イベント1つに対して1つのシナリオフロー図とするとしていたが、このようにした場合、機能イベントから機能イベントへの情報や、機能イベント以外のイベントとの情報の引き継

ぎをどのようにドキュメントに記述するか問題となった。

解決策の検討

シナリオフロー図の記述単位として、他に1画面に対して1つのシナリオフロー図を記述する方法が考えられる。しかし、同様の機能イベントが複数の画面で起こる場合、このような記述単位にすると、同じ機能が複数回使用されるため、同じシナリオフロー図を何度も記述しなければならない。従って、シナリオフロー図の記述単位については、当初の方法論どおり、1機能イベントに対して1つのシナリオフロー図を記述する事とした。

しかし、シナリオフロー図を機能イベント単位で記述する場合、現在の状況では、機能イベントから機能イベントへの情報や、機能イベント以外のイベントとの情報の引き継ぎを記述するドキュメントが存在しない。従って、これらの内容をドキュメントに記述するためのルールが必要となる。これは課題として、次の製造実験までに検討する事とした。

具体例1：画面設計図とシナリオフロー図のつながりを示すドキュメント

方法論では、基本設計工程において画面設計を行い、3層設計工程でシナリオフロー図を作成する。この画面設計図と、シナリオフロー図のつながりを示すドキュメントが必要である。これには、

- 画面内にどのような機能イベントやその他のイベントが存在するか
- 各イベントに対する入出力は何か
- 各機能イベントによって、どのシナリオフロー図の機能が実行されるか
- 各イベントによって、どのようにシステムの状態が遷移するか

の情報が必要がある。現在、このドキュメントに関して、内容と記述フォーマットについて検討中である。

問題点2 D層とF層の切り分けの基準はどうするか

D層にはデータを蓄積し、参照するための機能を配置するが、データを加工する必要がある場合、D層に配置すべき加工機能はどこまでであり、どこからの機能をF層に配置すべきかの基準が明らかでない。D層とF層の分割に関して明確な判断基準が存在しない場合、分担して作業する担当者が、各自の判断基準で分割し、インタフェースを設計する。このように設計されたインタフェースに関してインタフェース共通化を考えた場合、分割の基準が統一されていないため、共通化できる対象が少なくなり、インタフェース共通化によるシステム規模の縮小が期待できない可能性が考えられる。

解決策の検討

各層の機能の切り分けに関し、方法論ではその際の指針を与え、最終的な決定は業務の特性やプラットフォームの制約などを考慮し、開発者が判断するとしている。F層とD層の切り分けに関しても、汎用的な基準は存在しないと考える。今回実験で検討した、F層とD層機能の切り分けの指針に関しての具体例を下記に示す。

具体例2：D層、F層機能切り分けの設計指針

D層処理では、その機能がアクセスする実体について考えると、一般的に挿入、更新、削除、照会機能、また、複数件の挿入、更新、削除、照会機能が考えられる。従って、まずD層機能を設計しようとする際には、D層でこれらの機能を実現し、それ以上のデータの加工はF層で実現するように検討する。その後、全体を見渡し、同じようなデータの加工が存在する場合は、その機能をD層に含める事を検討する。

問題点3 機能の共通化をどのような基準で行うか

同じ様な機能を共通化する事により、全体の開発規模を縮小する事ができる。しかし、機能設計を行う際に設計者は汎用性を考え、他の機能と共通化しようとし、本来必要な機能より拡張して設計する。この際に拡張しすぎると無駄な開発を行う事になり、かえって開発規模が大きくなってしまふ可能性があ

る。

解決策の検討

共通化に関しても、機能の切り分けの問題と同様に、汎用的な基準は存在しないと考えられる。それぞれの開発プロジェクトにおいて、あらかじめ標準化作業を行い、ドメイン特性を考慮した共通化ルールを定める必要がある事が、今回の実験で明らかになった。方法論に、「機能共通化ルールの標準化」についての記述がなかったので、方法論に追加する事とした。予備実験で標準化した共通化ルールを具体例として下記に示す。

具体例3：共通化基準

1. D層の共通化を基準に考える。
2. D層とF層の分割問題と同様に、データを中心に考えた場合、1件だけの情報の登録、更新、削除、照会、および、複数件の登録、更新、削除、照会が考えられるため、D層の機能は、「1件」と「複数件」に分類する事にする。すなわち、「5件同時登録」と「10件同時登録」の機能が必要な場合、2つの機能を作成するのではなく、「複数件同時登録」として共通化する。しかし、別に「1件登録」が必要な場合には、共通化せずに、別機能として作成する。
3. F層の共通化はD層と同様に、D層から出力されるデータ中心に考える。
4. P層に関しては、共通化できそうな機能が少ないため、共通化は行わない事とする。

5.2.ドキュメントの評価

3層C/Sシステム設計方法論で作成するドキュメントについて、従来の開発手法で行った場合で作成するドキュメントとの違いについて考える。3層C/Sシステム設計方法論では、従来の設計方法論には存在しない、シナリオフロー図を作成する点が大きな特徴である。以下では、このシナリオフロー図を中心に評価する。

ドキュメントが見やすさ

従来のドキュメントでは、各機能ごとに仕様が記述され、処理の流れがわかりにくかったが、シナリオフロー図では、1つのシナリオフロー図で、1つの機能ごとに処理の流れが、ユーザインタフェースからデータベースまで通して見る事ができる。予備実験で実際にシナリオフロー図を記述した結果、処理の流れが理解しやすい事が確認できた。

ドキュメントの課題

予備実験では、約3Kラインのシステムで、シナリオフロー図を約50枚作成した。この比率で考えると、実際システムで1Mライン程度の規模であれば、約16,000枚のシナリオフロー図を作成する事になる。実際にはこのように多量のシナリオフロー図を作成する事は困難であるため、今後枚数を削減する方法を検討する必要がある。

5.3.考察

今回の予備実験では、机上の検討では、検出することができなかったいくつかの問題点が明らかになった。シナリオフロー図に関して、予備実験では、具体例1のような方針で作成し問題はなかったが、ドキュメントの課題にあげたような量の問題があるので、今後記述レベルなどに関して詳細に検討する必要がある。問題点2と問題点3については、一般的な解答は存在しない。開発者がプロジェクトのドメイン特性等を考慮して方針を決定する必要がある。このような問題に関して、開発者が方針をすみやかに決定できるように、方法論においては、具体例に基づいた指針を記述する事とした。開発工数の指標値に関しては、今回の実験が初めてであるため、ツールを習熟するための時間や、環境に慣れるための時間が含まれているため、正確な値が得られないと考えられる。指標値に関しては、今後より厳密な実験を重ねて検討する必要がある。

6.今後の評価実験の進め方

今回予備実験を行った結果をもとに、次の製造実験を以下のように行う。

- 製造実験は、予備実験の結果を反映し、改訂された方法論ののっとり実験を進める。製造実験では、実際にシステム開発を行う開発グループと、方法論を検討し、開発グループを観測する観測者グループにわかれて行う。開発グループのメンバーは、方法論の内容に対して先入観のない、予備実験に参加した以外のメンバーで構成し、予備実験に参加したメンバーは、観測者グループとして開発グループを管理し、方法論の詳細な評価検討を行う。
- 製造実験の実験計画は、予備実験で得られた指標値をもとに見積もり、製造実験からもより厳密な指標値を抽出する。
- 製造実験の目的は、詳細な方法論の評価と、より厳密な指標値を抽出する事とする。

7.おわりに

本稿では、3層C/Sシステム開発方法論の有効性と問題点を明らかにするために実施した評価実験のうち、予備実験の結果について報告した。予備実験では、以下のような事が明らかになった。

- シナリオフロー図作成の単位は、機能イベント単位とした場合、小規模なシステムでは問題ないが、大規模システムでは量が多くなるため、今後記述レベルなど、量を削減するための検討が必要である。
- D層・F層の機能切り分けに関する、具体的な指針を検討した。
- 機能の共通化に関しては、開発プロジェクトごとの標準化が必要であり、標準化の為の指針を検討した。
- 3層C/Sシステム設計方法論で作成されるドキュメントに関して評価を行った。
- 平均5個の機能イベントを有する画面1つを

開発するための開発工数は、約10人日であるとの指標値が得られた。ただし、今回が初めての開発であったため、この値にはツールに習熟するための時間や、環境になれるための時間などが含まれているため、今後より厳密な指標値の抽出が必要である。

謝辞

本評価実験にご協力いただいた、コムニク創研の皆様、有益なコメントをしていただいた、ソフトウェア技術研究部の高木 浩則君、村田 博史君、ならびに日頃ご指導いただき、細谷 僚一ソフトウェア研究所所長および長野 宏宣ソフトウェア技術研究部長に深謝いたします。

参考文献

- [1] ジョン・J・ドノヴァン著、「情報技術とビジネス・リエンジニアリング」、産能大学出版部刊
- [2] 高田 信一、畑 恵介、山本 修一郎、「3層クライアント・サーバアーキテクチャに適應したシステム設計手法の提案」、第56回情報システム研究会、1995.10.17
- [3] 高田 信一、畑 恵介、山本 修一郎、「3層クライアント/サーバ設計法の適用事例」、第51情報処理学会全国大会、4N-3、1995.9
- [4] 加藤 謹詞、元田 敏浩、黒川 裕彦、長岡 満夫、「分散OA構築支援システムVGUIDEのオープンシステム化」、NIT R&D、Vol.41 No.12 1992
- [5] 長岡 満夫、「エンドユーザコンピューティング技術」、NIT R&D、Vol.42 No.12 1993
- [6] 拜原 正人、「クライアント/サーバによる分散処理の動向(その2)、クライアント/サーバの実際と今後」、NIT 技術ジャーナル、1994年12月